Kevin Romero Peces-Barba
kromero8@ucsc.edu
1635745

CMPS 142 - Spring 2018

Homework 1 - Problem 6

In this question, you will learn how to use a Machine Learning toolkit, Weka. Specifically, we will be learn De- cision Trees for predicting survival using the Titanic dataset. It contains personal information about passengers and their tickets along with whether they survived. We have already split the data into train and test files: cmps142 hw1 train.csv and cmps142 hw1 test.csv. If interested, you could learn more about the corpus on its Description page. You need to learn using Weka (3.8.2) to train a Decision Tree classifier. The download page of Weka is here. A tutorial of how to use Weka to train a decision tree classifier is here. For this problem, we treat the 'survived' column of the corpus as label to be predicted.

1. Preprocessing: The corpus has several attributes. By default, Weka assumes that most of them are numeric. However, some of the attributes would be meaningful as categorical/ nominal attributes. In the first step, you will use Weka's inbuilt filter to convert these attributes from numeric to nominal for both the train and the test files. Specifically, you will use the filter named NumericToNominal to convert the following attributes to nominal: Pclass, Parch, Has Cabin, IsAlone, Survived. After you have converted the above mentioned attributes to nominals for both the given files, save the resulting files as arff files named cmps142 hw1 train.arff and cmps142 hw1 test.arff respectively. Note that this can be done on the Weka GUI. You can read more about the arff format here.
   What to submit: You have to submit the two arff files with your report.
   Questions to answer in report: Using the visualization tool in Weka's GUI answer the following questions
   (a) How many unique values can the Parch attribute take in the train set? What are they?

   There's only one unique value: Parch=6.

   (b) How many unique values can the Parch attribute take in the test set? What are they?

   There's only one unique value: Parch=5¶

(c) Note that Weka automatically recognizes Sex, Embarked and Title as Nominal attributes. Why does Weka think that an attribute like Embarked is nominal while an attribute like Parch isn't?

Sex, Embarked and Title all have values that are alphabetic strings, while parch has values that are numbers. This is why weka thinks parch is numeric and not nominal, and why it classifies embarked as nominal. This same thing happens for every boolean value labeled with {0, 1}.

2. Building a tree: For the rest of this homework, you will only work with the preprocessed arff files that you created in the previous step. Use C4.5 (J48) Decision Tree algorithm with Weka's default settings to learn a Decision Tree classifier on the training set. Note that the default settings are: C=0.25, M=2, and unpruned=False. For this homework assignment we are only concerned with these 3 parameters. The rest of the parameters/options should not be changed. We will be using 10-fold Cross-validation (CV) on the train set to evaluate the learned decision tree.
Questions to answer in report: Answer the following questions using Weka's GUI.
(a) What is the 10-fold CV accuracy?

Accuracy is 551/701 = 78.602%.

(b) What is the confusion matrix for your 10-fold CV?

	a	b
a	364	65
b	85	187

(c) What is the (i) number of leaves and (ii) tree size as reported by Weka?

Number of leaves is 2 and tree size is 3.

3. Pruning: The tree built in the previous question was pruned. Now you will build an unpruned tree and analyze the difference in performance. To train an unpruned tree, use C4.5 (J48) Decision Tree algorithm with the following settings: M=2, and unpruned=True (setting unpruned=True renders C inactive). As before, report 10-fold CV performance on the train set.
Questions to answer in report:
(a) What is the 10-fold CV accuracy?

The accuracy is 566/701 = 80.7418%.¶

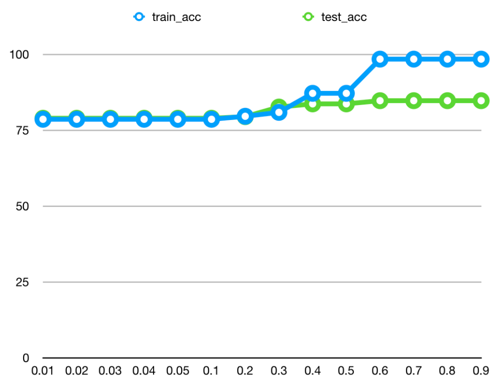
(b) What is the (i) number of leaves and (ii) tree size as reported by Weka?

Number of leaves is 1103 and tree size is 1117.

(c) How does the performance of the unpruned method compare with the performance using pruning? Also, give a reason for your observation.

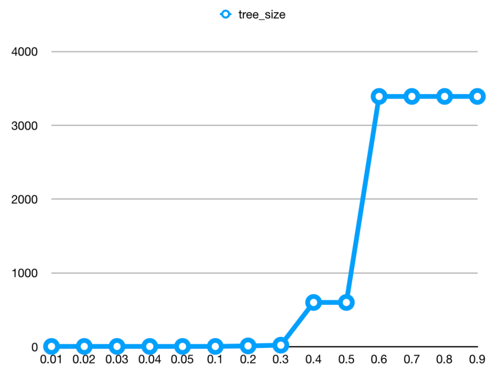
Unpruned trees have no tree size/branching limit, so the size and density of the tree increase a lot, causing the performance to lower.

4. Effect of pruning: For this question, you will be training the tree using the train set and reporting performance on the training set itself as well as separately provided test set. Note that Weka allows you to do both. Weka's implementation of the algorithm has a parameter, C confidenceFactor, which controls the degree to which the tree is pruned. Smaller values of C result in more pruned trees. In this experiment, you will see the change in training and test performance with changing values of this parameter. Also, the value of M should be set to be 0.
Questions to answer in report:
(a) Draw a plot of C versus training and test accuracies. The x-axis should report the following values of C {0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}, and the y-axis should be training (and test accuracies). Report the trend that you observe. Why do you see this trend? Do not forget to include the plot in your submission.



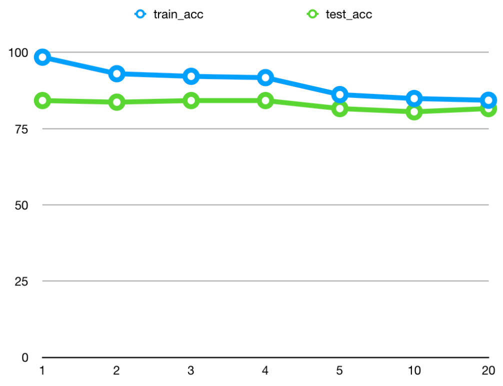
We can't really see the effect until the confidenceFactor reaches a certain point. Larger values of C allow for bigger (less pruned) trees. This way the overall accuracy of the model increases.
However, if larger trees are allowed, the model overfits to the train data while not getting better results in unseen data.
There's also a point where the algorithm won't create better trees.

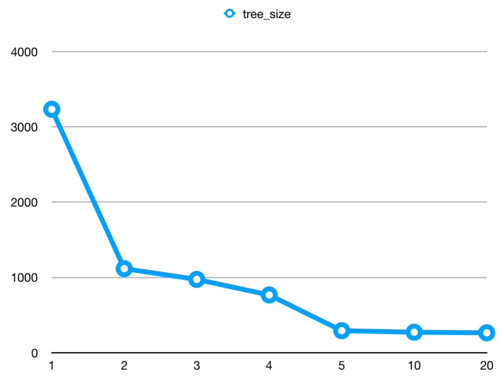
(b) Draw a plot of C versus size of tree. The x-axis should report the following values of C {0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}, and the y-axis should be the corresponding tree size. Report the trend that you observe. Why do you see this trend? Do not forget to include the plot in your submission.



As C controls exactly the amount of pruning, increasing it results in less pruned trees. That's why we get bigger trees.
However, this doesn't happen until a certain amount of confidence is reached and, from this point, the size grows exponentially until it gets stuck again because no better (ie, more complex) trees can be found.

5. This question is similar to the previous one. In this question, you will study the effect of varying M on the performance on the train and the test sets. Plot the same graphs as above while trying the following values of M: {1,2,3,4,5,10,20}. Set unpruned=True. Report the trends you observe and an explanation for the trends. Do not forget to include the plots in your submission.





This parameter controls the minimum number of instances that J48 should have, at least, in the two most popular branches, for each split. This basically means that it won't select a certain split unless it ensures that the two branches with more data have at least M leafs.

As we increase this parameter, the size of the tree heavily decreases, because less splits are taken (until it reaches a stability point)
However, we can see that the accuracy in unseen data does not decrease, because better splits are taken. Moreover, this also makes the model overfit less the training data.